

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.



Welcome
United States Patent and Trademark Office



Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet



Print Format

Your search matched **0** of **1074479** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or entering a new one in the text box.

universal <near> resource <near> identifier<and>jav

Search

☐ Check to search within this result set

Results Key:

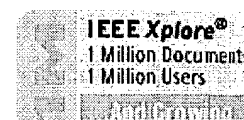
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:

No documents matched your query.



Welcome
United States Patent and Trademark Office



[Help](#) [FAQ](#) [Terms](#) [IEEE Peer Review](#)

[Quick Links](#)

» [Search Results](#)

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet

 [Print Format](#)

Your search matched **21** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 Interictal spike detection using the Walsh transform

Adjouadi, M.; Sanchez, D.; Cabrerizo, M.; Ayala, M.; Jayakar, P.; Yaylali, I.; Barreto, A.;

Biomedical Engineering, IEEE Transactions on , Volume: 51 , Issue: 5 , May 2004
Pages:868 - 872

[\[Abstract\]](#) [\[PDF Full-Text \(224KB\)\]](#) **IEEE JNL**

2 Approximation-in-policy-space for periodic optimization

Dorato, P.; Sanchez, D.; Yuan-Shyon Wang;

Automatic Control, IEEE Transactions on , Volume: 28 , Issue: 4 , Apr 1983
Pages:529 - 531

[\[Abstract\]](#) [\[PDF Full-Text \(280KB\)\]](#) **IEEE JNL**

3 Fuzzy association rules: general model and applications

Delgado, M.; Marin, N.; Sanchez, D.; Vila, M.-A.;

Fuzzy Systems, IEEE Transactions on , Volume: 11 , Issue: 2 , April 2003
Pages:214 - 225

[\[Abstract\]](#) [\[PDF Full-Text \(717KB\)\]](#) **IEEE JNL**

4 Voltage balance and control in a multi-level unified power flow controller

Soto-Sanchez, D.E.; Green, T.C.;

Power Delivery, IEEE Transactions on , Volume: 16 , Issue: 4 , Oct. 2001
Pages:732 - 738

[\[Abstract\]](#) [\[PDF Full-Text \(152KB\)\]](#) **IEEE JNL**

5 A library for load balancing in master/slave applications on a LAN-WLAN environment

Sanchez, D.; Macias, E.M.; Suarez, A.;

Parallel, Distributed and Network-Based Processing, 2004. Proceedings. 12th Euromicro Conference on , 11-13 Feb. 2004
Pages:168 - 175

[\[Abstract\]](#) [\[PDF Full-Text \(298KB\)\]](#) **IEEE CNF**

6 Orthogonalized algorithms for FIR ANC adaptive systems

Tapia-Sanchez, D.; Bello, R.B.; Perez-Meana, H.; Nakano-Miyatake, M.;

Antenna Theory and Techniques, 2003. IVth International Conference on , Volume: 1 , 9-12 Sept. 2003
Pages:337 - 340 vol.1

[\[Abstract\]](#) [\[PDF Full-Text \(296KB\)\]](#) [IEEE CNF](#)

7 Pipelining-based tradeoffs for hardware/software codesign of multimedia systems

Castellano, J.P.; Sanchez, D.; Cazorla, O.; Suarez, A.;
Parallel and Distributed Processing, 2000. Proceedings. 8th Euromicro Workshop on , 19-21 Jan. 2000
Pages:383 - 390

[\[Abstract\]](#) [\[PDF Full-Text \(88KB\)\]](#) [IEEE CNF](#)

8 Fuzzy genes: improving the effectiveness of information retrieval

Martin-Bautista, M.J.; Vila, M.-A.; Sanchez, D.; Larsen, H.L.;
Evolutionary Computation, 2000. Proceedings of the 2000 Congress on , Volume: 1 , 16-19 July 2000
Pages:471 - 478 vol.1

[\[Abstract\]](#) [\[PDF Full-Text \(556KB\)\]](#) [IEEE CNF](#)

9 Integrating proprietary managed PDH networks using TMN-based platforms

Guerrero, C.; Sanchez, D.; Carneiro, V.; Vina, A.; Coego, J.;
Enterprise Networking Mini-Conference, 1997. ENM-97. In conjunction with the ICC-97., First IEEE , 11-12 June 1997
Pages:180 - 190

[\[Abstract\]](#) [\[PDF Full-Text \(792KB\)\]](#) [IEEE CNF](#)

10 Definition and implementation of a TMN information model for legacy PDH equipments

Sanchez, D.; Guerrero, C.; Vina, A.;
EUROMICRO 97. 'New Frontiers of Information Technology'. Short Contributions., Proceedings of the 23rd Euromicro Conference , 1-4 Sept. 1997
Pages:82 - 87

[\[Abstract\]](#) [\[PDF Full-Text \(340KB\)\]](#) [IEEE CNF](#)

11 Minimizing buffer requirements in video-on-demand servers

Garcia-Martinez, A.; Molano, A.; Sanchez, D.; Vina, A.;
EUROMICRO 97. 'New Frontiers of Information Technology'. Short Contributions., Proceedings of the 23rd Euromicro Conference , 1-4 Sept. 1997
Pages:68 - 73

[\[Abstract\]](#) [\[PDF Full-Text \(360KB\)\]](#) [IEEE CNF](#)

12 Drum roll: increasing bandwidth through passive impedance modulation

Hajian, A.Z.; Sanchez, D.S.; Howe, R.D.;
Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on , Volume: 3 , 7-11 Sept. 1997
Pages:V35 - V36 vol.3

[\[Abstract\]](#) [\[PDF Full-Text \(320KB\)\]](#) [IEEE CNF](#)

13 Drum roll: increasing bandwidth through passive impedance modulation

Hajian, A.Z.; Sanchez, D.S.; Howe, R.D.;

Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on , Volume: 3 , 20-25 April 1997

Pages:2294 - 2299 vol.3

[\[Abstract\]](#) [\[PDF Full-Text \(668KB\)\]](#) [IEEE CNF](#)

14 A hierarchical approach to fuzzy segmentation of colour images

Chamorro-Martinez, J.; Sanchez, D.; Prados-Suarez, B.; Galan-Perales, E.; Vila, M.A.;

Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on , Volume: 2 , 25-28 May 2003

Pages:966 - 971 vol.2

[\[Abstract\]](#) [\[PDF Full-Text \(452KB\)\]](#) [IEEE CNF](#)

15 Visualization for cooperative architecture design system

Yuhua Luo; Sanchez, D.; Bennasar, A.; Fornes, J.; Serra, J.C.; Huescar, J.M.;

Information Visualisation, 2002. Proceedings. Sixth International Conference on , 10-12 July 2002

Pages:497 - 501

[\[Abstract\]](#) [\[PDF Full-Text \(564KB\)\]](#) [IEEE CNF](#)

[1](#) [2](#) [Next](#)



[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

"rajesh peddu"

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 3 of about 4 for "rajesh peddu". (0.36 second)

Tip: Try removing quotes from your search to get more results.

IBM Tivoli Lotus "Run for the Children" Open

... Arun Deivanayagam Austin TX 25 123 24:23.3 24:35.5 0:12.2 7:52/M 17 Todd Muirhead
Round Rock TX 27 129 24:49.6 25:14.9 0:25.3 8:00/M 18 **Rajesh Peddu** Austin TX ...
209.198.135.156/ibmdownloads/ibm%20age%20grp.htm - 61k - [Cached](#) - [Similar pages](#)

IBM Tivoli Lotus "Run for the Children"

... 8:02/M 171 Oscar Cepeda Austin TX 35 24:59.1 26:01.7 1:02.6 8:04/M 172 Dominique
Torres Austin TX 25 25:00.3 25:09.5 0:09.2 8:04/M 173 **Rajesh Peddu** Austin TX ...
209.198.135.156/ibmdownloads/IBM%20OVERALL.HTM - 50k - [Cached](#) - [Similar pages](#)

DAEEE Committee

... Treasurer : CH.VLN Naveen kumar. CHIEF ORGANISING COMMITTEE: **Rajesh Peddu**.
Rajesh Pentapati. J.Aditya Sai. KVS.Prasad. SNS.Pavan Kiran. ...
members.fortunecity.com/gitambulls/id16_m.htm - 26k - Supplemental Result - [Cached](#) - [Similar pages](#)

In order to show you the most relevant results, we have omitted some entries very similar to the 3 already displayed.

If you like, you can repeat the search with the omitted results included.

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)



"rajesh peddu"

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied?](#) [Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google



[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

"david eugene sanchez"

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 1 of 1 for "**david eugene sanchez**". (0.51 second)

Tip: Try removing quotes from your search to get more results.

LAS VEGAS REVIEW-JOURNAL: NEWS: Deaths

... Memorial services will be at 10 am Tuesday at Christ The King Catholic Church. David Sanchez. **David Eugene Sanchez**, 50, died Sept. 21. He was born Aug. ...

www.reviewjournal.com/lvrj_home/2002/Oct-06-Sun-2002/news/19781922.html - 32k - [Cached](#) - [Similar pages](#)

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)



"david eugene sanchez"

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google



[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

"david e. sanchez"

[Search](#)

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 28 for "**david e. sanchez**". (0.42 seconds)

[\[PDF\] CITY COUNCIL MEETING OF](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... 2004-39 was adopted approving the establishment of two (2) boulevard stops on Railroad Avenue for the **David E. Sanchez** Elementary School. Resolution No. ...

www.ci.santa-maria.ca.us/minutes/CityCouncil-20040316RM.pdf - [Similar pages](#)

[The Reveille Club - Job Opportunities](#)

... 2004. Adaptable and Optimistic Manufacturing Eng. **David E. Sanchez**, 08/04/2004. Human Resources Professional, Sandy Rosner, 08/03/2004. ...

www.reveille.org/php/viewresumes.php - 101k - [Cached](#) - [Similar pages](#)

[The Actinide Research Quarterly: Spring 1996](#)

... instrument. D. Kirk Veirs, Noah G. Pope, **David E. Sanchez**, Vincente D. Sandoval are the developers of this diagnostic technique. ...

www.lanl.gov/orgs/nmt/nmtdo/AQarchive/95summer/NCCA.html - 10k - [Cached](#) - [Similar pages](#)

[\[PDF\] The Actinide Research Quarterly](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

Page 1. 1 Nuclear Materials Technology Division/Los Alamos National Laboratory Summer 1995 Los Alamos National Laboratory • A US ...

www.lanl.gov/orgs/nmt/nmtdo/AQarchive/95summer/95summer.pdf - [Similar pages](#)

[\[PDF\] Pulse - Summer/Fall 02\(first 2\)](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

Page 1. 52 Pulse / Spring 2003 Memorials Memorial gifts provide a way for people to express their sympathy when words just don't seem adequate. ...

www.chomp.org/pulse/pages/2002_memorials_tributes.pdf - [Similar pages](#)

[Donor List](#)

... Mr. and Mrs. Fred Salyer Mr. Lee T. Samore San Benancio School San Benito County Health and Human Services Employees Mr. and Mrs. **David E. Sanchez** Mr. and Mrs ...

www.chomp.org/pulse/pages/2002_donors_healthcare.html - 91k - [Cached](#) - [Similar pages](#)

[Re: SS Willesden - information wanted](#)

... In Reply to: Re: SS Willesden - information wanted posted by **david e sanchez** on February 04, 2003 at 02:10:53: Mr. **David E. Sanchez** --. ...

www.steamboat.org.uk/wwwboard/messages/4509.htm - 5k - [Cached](#) - [Similar pages](#)

[Re: SS Willesden - information wanted](#)

... Posted by **david e sanchez** on February 04, 2003 at 02:10:53: In Reply to: SS Willesden - information wanted posted by D Nilssen on March 21, 2002 at 10:02:04: ...

www.steamboat.org.uk/wwwboard/messages/4042.htm - 4k - [Cached](#) - [Similar pages](#)

[[More results from www.steamboat.org.uk](#)]

[Nervousness :: Forums > Bride of the Super Postcard Swap](#)

... Aug 31 2003, 02:17 PM. #11 **David E Sanchez** AKA Spooker <-- my name PO Box 1255 Jefferson, Oregon 97352 <-- my address sirspookyspooker@aol.com <-- my email. ...

forums.nervousness.org/lofiversion/index.php/t5215.html - 25k - [Cached](#) - [Similar pages](#)

[TurnKey - Site Map](#)

... School; **David E. Sanchez** Elementary School; Ernest R. Geddes Elementary School; Santa Maria-Bonita School District Gymnasiums; Woodrow ...

www.turnkey.bz/sitemap.htm - 18k - [Cached](#) - [Similar pages](#)

Google 

Result Page: 1 [2](#) [3](#) [Next](#)

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)



[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google



Web Images Groups News Froogle [more »](#)

+content +template +java +server +page

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 322,000 for +content +template +java +server +page. (0.40 second)

Servlet Tutorial: JavaServer Pages (JSP) 1.0

... the following JSP fragment, containing mixed **template** text and ... CENTER> <P> Some dynamic **content** created using ... B>
 <%@ **page** import = "java.util.*" %> Current ...
www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-JSP.html - 49k -
[Cached](#) - [Similar pages](#)

Implementing Conditional **Content** on a JSP Page (Java Developers ...

... examples] > **Java Server** Pages Output [4 examples]. e1059. Implementing Conditional **Content** on a JSP Page. Any non-JSP code in a JSP file is called **template** text ...
javaalmanac.com/egs/javax.servlet.jsp/if.html - 6k - [Cached](#) - [Similar pages](#)

Servlets.com | The Problems with JSP

... JSP to another technology built on servlets: **template** engines ... to ask developers to make all **content** changes ... technology you could put snippets of **Java** code inside ...
www.servlets.com/soapbox/problems-jsp.html - 36k - Sep 22, 2004 -
[Cached](#) - [Similar pages](#)

jGuru: JavaServer Pages Fundamentals, Short Course Contents

... Dynamic **content** can be served in a variety of ... nothing that mandates the static **template** data within ... applications using the enterprise **Java** APIs, categorically ...
developer.java.sun.com/developer/onlineTraining/JSPIntro/contents.html - 54k -
[Cached](#) - [Similar pages](#)

jGuru: JSP FAQ Home Page

... the separation of presentation from dynamic web **content**. ... using <jsp:include **page**="/access/**template**.jsp" flush ... their username and password, which i... **Java**:API: ...
www.jguru.com/faq/JSP - 31k - Sep 22, 2004 - [Cached](#) - [Similar pages](#)

Java Server Pages

Java Server Pages are an HTML **template** technology similar in ... **Page** is HTML with embedded **Java** and other ... a relatively small amount of dynamic **content** is embedded ...
c2.com/cgi/wiki?JavaServerPages - 13k - [Cached](#) - [Similar pages](#)

SwTech.com - Java Servlets and Server-side Java

... such as on-the-fly translation, **content** filters, or ... to use simple yet powerful **template** language to ... providing a viable alternative to **Java Server** Pages (JSPs ...
www.swtech.com/java/servlet/ - 40k - [Cached](#) - [Similar pages](#)

How to use FrontPage to create accessible **content**

... 3.1 Create accessible templates and add **content** within them. ... that are not a part of the generic **template**. ... use scripting languages such as **Java Server** Pages, PHP ...
www.webaim.org/techniques/frontpage/ - 36k - Sep 22, 2004 - [Cached](#) - [Similar pages](#)

What's New

... Stellent **Content Server** core functionality can be accessed from a **Java Server Page** through the ... The **Content Server Workflow Template** area contains a ...
www.ogc1.umn.edu/stellent/help/whatsnew.htm - 38k - [Cached](#) - [Similar pages](#)

Java Server Programming Tag Extensions Part 4 - Tag Extension ...

... whole development community, whereas your tag **content** may not ... Consider reading the HTML from a **template** file ... require plunging into the tag's **Java** implementation ...
tutorials.findtutorials.com/read/id/234 - 49k - [Cached](#) - [Similar pages](#)

Sponsored Links

[Web Content Management](#)
User-friendly and flexible **Java** solution. Fast, easy and affordable
www.enonic.com

[See your message here...](#)

Google

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#)



+content +template +java +serve **Search**

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied?](#) [Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google

JavaServer Pages (JSP) 1.0

1. [Overview](#)
2. [Syntax Summary](#)
3. [Template Text \(Static HTML\)](#)
4. [JSP Scripting Elements: Expressions, Scriptlets, and Declarations](#)
5. [JSP Directives](#)
6. [Example using Scripting Elements and Directives](#)
7. [Predefined Variables](#)
8. [JSP Actions](#)
 - o [8.1. jsp:include](#)
 - o [8.2. jsp:useBean overview](#)
 - o [8.3. jsp:useBean details](#)
 - o [8.4. jsp:setProperty](#)
 - o [8.5. jsp:getProperty](#)
 - o [8.6. jsp:forward](#)
 - o [8.7. jsp:plugin](#)
9. [JSP Comments and Character Escaping Conventions](#)
10. [Servlet and JSP Tutorial: Top](#)
11. [Servlet and JSP Training Courses On-site at *your* company or at public venues.](#)

Tomcat Hosting Services

Tomcat 5 & 4, PostgreSQL, MySQL 4, RedHat 9, Private JVM from \$13.95

5GB Web Space - \$3.95/mth

99.5% Uptime & Toll Free Support 30 Day Money Back Guarantee

Ads by Goooooogle

Note: [Core Servlets and JavaServer Pages](#) contains much more detailed and up to date information on this topic. The full text is now [available online in PDF](#) for free access.

1. Overview

JavaServer Pages (JSP) lets you separate the dynamic part of your pages from the static HTML. You simply write the regular HTML in the normal manner, using whatever Web-page-building tools you normally use. You then enclose the code for the dynamic parts in special tags, most of which start with "<%" and end with "%>". For example, here is a section of a JSP page that results in something like "Thanks for ordering *Core Web Programming*" for a URL of <http://host/OrderConfirmation.jsp?title=Core+Web+Programming>:

```
Thanks for ordering
<I><%= request.getParameter("title") %></I>
```

You normally give your file a .jsp extension, and typically install it in any place you could place a normal Web page. Although what you write often looks more like a regular HTML file than a servlet, behind the scenes, the JSP page gets converted to a normal servlet, with the static HTML simply being printed to the output stream associated with the servlet's `service` method. This is normally done the first time the page is requested, and developers can simply request the page themselves when first installing it if they want to be sure that the first real user doesn't get a momentary delay when the JSP page is translated to a servlet and the servlet is compiled and loaded. Note also that many Web servers let you define aliases so that a URL that appears to reference an HTML file really points to a servlet or JSP page.

Aside from the regular HTML, there are three main types of JSP constructs that you embed in a page: scripting elements, directives, and actions. *Scripting elements* let you specify Java code that will become part of the resultant servlet, *directives* let you control the overall structure of the servlet, and *actions* let you specify existing components that should be used, and otherwise control the behavior of the JSP engine. To simplify the scripting elements, you have access to a number of predefined variables such as `request` in the snippet above.

Note that this tutorial covers version 1.0 of the JSP specification. JSP has changed dramatically since version 0.92, although these changes were almost entirely for the better, you should note that version 1.0 JSP pages are almost totally incompatible with the earlier JSP engines. Note that this JSP tutorial is part of a larger tutorial on servlets and JSP at <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>.

2. Syntax Summary

JSP Element	Syntax	Interpretation	Notes
JSP Expression	<code><%= expression %></code>	Expression is evaluated and placed in output.	XML equivalent is <code><jsp:expression>expression</jsp:expression></code> . Parameters are request, response, application, config, a pageContext (available in
JSP Scriptlet	<code><% code %></code>	Code is inserted in service method.	XML equivalent is <code><jsp:scriptlet>code</jsp:scriptlet></code> .
JSP Declaration	<code><%! code %></code>	Code is inserted in body of servlet class, outside of service method.	XML equivalent is <code><jsp:declaration>code</jsp:declaration></code> .
JSP page Directive	<code><%@ page att="val" %></code>	Directions to the servlet engine about general setup.	XML equivalent is <code><jsp:directive.page</code> Legal attributes, with default values: <ul style="list-style-type: none"> • import="<i>package.class</i>" • contentType="<i>MIME-Type</i>" • isThreadSafe="true false" • session="true false" • buffer="<i>sizekb</i> none" • autoFlush="true false" • extends="<i>package.class</i>" • info="<i>message</i>" • errorPage="<i>url</i>" • isErrorPage="true false" • language="<i>java</i>"
JSP include Directive	<code><%@ include file="url" %></code>	A file on the local system to be included when the JSP page is translated into a servlet.	XML equivalent is <code><jsp:directive.incl file="url"></code> . The URL must be a relative or absolute URL. The <code>jsp:include</code> action to include a file at request time instead of translation time.
JSP Comment	<code><!-- comment --></code>	Comment; ignored when JSP page is translated into servlet.	If you want a comment in the output, use regular HTML comments: <code>comment --></code> .
The <code>jsp:include</code> Action	<code><jsp:include page="relative URL"</code>	Includes a file at the time the page is translated.	If you want to include the file at request time, use the <code>jsp:include</code> action. On some servers, the included file

	<code>flush="true"/></code>	requested.	HTML file or JSP file, as determined by the server (usually based on the file extension)
The <code>jsp:useBean</code> Action	<code><jsp:useBean att=val*/> or <jsp:useBean att=val*> ... </jsp:useBean></code>	Find or build a Java Bean.	Possible attributes are: <ul style="list-style-type: none"> • <code>id="name"</code> • <code>scope="page request session"</code> • <code>class="package.class"</code> • <code>type="package.class"</code> • <code>beanName="package.className"</code>
The <code>jsp:setProperty</code> Action	<code><jsp:setProperty att=val*/></code>	Set bean properties, either explicitly or by designating that value comes from a request parameter.	Legal attributes are <ul style="list-style-type: none"> • <code>name="beanName"</code> • <code>property="propertyName"</code> • <code>param="parameterName"</code> • <code>value="val"</code>
The <code>jsp:getProperty</code> Action	<code><jsp:getProperty name="propertyName" value="val"/></code>	Retrieve and output bean properties.	
The <code>jsp:forward</code> Action	<code><jsp:forward page="relative URL"/></code>	Forwards request to another page.	
The <code>jsp:plugin</code> Action	<code><jsp:plugin attribute="value"*> ... </jsp:plugin></code>	Generates OBJECT or EMBED tags, as appropriate to the browser type, asking that an applet be run using the Java Plugin.	

3. Template Text: Static HTML

In many cases, a large percent of your JSP page just consists of static HTML, known as *template text*. In all respects except one, this HTML looks just like normal HTML, follows all the same syntax rules, and is simply "passed through" to the client by the servlet created to handle the page. Not only does the HTML *look* normal, it can be *created* by whatever tools you already are using for building Web pages. For example, I used Allaire's HomeSite for most of the JSP pages in this tutorial.

The one minor exception to the "template text is passed straight through" rule is that, if you want to have "<%" in the output, you need to put "<\%" in the template text.

4. JSP Scripting Elements

JSP scripting elements let you insert Java code into the servlet that will be generated from the current JSP page. There are three forms:

1. Expressions of the form `<%= expression %>` that are evaluated and inserted into the output,
2. Scriptlets of the form `<% code %>` that are inserted into the servlet's `service` method, and
3. Declarations of the form `<%! code %>` that are inserted into the body of the servlet class, outside of any existing methods.

Each of these is described in more detail below.

4.1 JSP Expressions

A JSP *expression* is used to insert Java values directly into the output. It has the following form:

```
<%= Java Expression %>
```

The Java expression is evaluated, converted to a string, and inserted in the page. This evaluation is performed at run-time (when the page is requested), and thus has full access to information about the request. For example, the follow shows the date/time that the page was requested:

```
Current time: <%= new java.util.Date() %>
```

To simplify these expressions, there are a number of predefined variables that you can use. These implicit objects are discussed in more detail later, but for the purpose of expressions, the most important ones are:

- `request`, the `HttpServletRequest`;
- `response`, the `HttpServletResponse`;
- `session`, the `HttpSession` associated with the request (if any); and
- `out`, the `PrintWriter` (a buffered version of type `JspWriter`) used to send output to the client.

Here's an example:

```
Your hostname: <%= request.getRemoteHost() %>
```

Finally, note that XML authors can use an alternative syntax for JSP expressions:

```
<jsp:expression>  
Java Expression  
</jsp:expression>
```

Remember that XML elements, unlike HTML ones, are case sensitive. So be sure to use lowercase.

4.2 JSP Scriptlets

If you want to do something more complex than insert a simple expression, JSP *scriptlets* let you insert arbitrary code into the servlet method that will be built to generate the page. Scriptlets have the following form:

```
<% Java Code %>
```

Scriptlets have access to the same automatically defined variables as expressions. So, for example, if you want output to appear in the resultant page, you would use the `out` variable.

```
<%  
String queryData = request.getQueryString();  
out.println("Attached GET data: " + queryData);
```

%>

Note that code inside a scriptlet gets inserted *exactly* as written, and any static HTML (template text) before or after scriptlet gets converted to `print` statements. This means that scriptlets need not contain complete Java statements, and blocks left open can affect the static HTML outside of the scriptlets. For example, the following JSP fragment, containing mixed template text and scriptlets

```
<% if (Math.random() < 0.5) { %>
Have a <B>nice</B> day!
<% } else { %>
Have a <B>lousy</B> day!
<% } %>
```

will get converted to something like:

```
if (Math.random() < 0.5) {
    out.println("Have a <B>nice</B> day!");
} else {
    out.println("Have a <B>lousy</B> day!");
}
```

If you want to use the characters "%>" inside a scriptlet, enter "%\>" instead. Finally, note that the XML equivalent of `<% Code %>` is

```
<jsp:scriptlet>
Code
</jsp:scriptlet>
```

4.3 JSP Declarations

A JSP *declaration* lets you define methods or fields that get inserted into the main body of the servlet class (outside of the `service` method processing the request). It has the following form:

```
<%! Java Code %>
```

Since declarations do not generate any output, they are normally used in conjunction with JSP expressions or scriptlets. For example, here is a JSP fragment that prints out the number of times the current page has been requested since the server booted (or the servlet class was changed and reloaded):

```
<%! private int accessCount = 0; %>
Accesses to page since server reboot:
<%= ++accessCount %>
```

As with scriptlets, if you want to use the characters "%>", enter "%\>" instead. Finally, note that the XML equivalent of `<%! Code %>` is

```
<jsp:declaration>
Code
</jsp:declaration>
```

5. JSP Directives

A JSP *directive* affects the overall structure of the servlet class. It usually has the following form:

```
<%@ directive attribute="value" %>
```

However, you can also combine multiple attribute settings for a single directive, as follows:

```
<%@ directive attribute1="value1"
      attribute2="value2"
      ...
      attributeN="valueN" %>
```

There are two main types of directive: `page`, which lets you do things like import classes, customize the servlet superclass, and the like; and `include`, which lets you insert a file into the servlet class at the time the JSP file is translated into a servlet. The specification also mentions the `taglib` directive, which is not supported in JSP version 1.0, but is intended to let JSP authors define their own tags. It is expected that this will be the main new contribution JSP 1.1.

5.1 The JSP page Directive

The `page` directive lets you define one or more of the following case-sensitive attributes:

- `import="package.class"` or `import="package.class1,...,package.classN"`. This lets you specify what packages should be imported. For example:

```
<%@ page import="java.util.*" %>
```

 The `import` attribute is the only one that is allowed to appear multiple times.
- `contentType="MIME-Type"` or
`contentType="MIME-Type; charset=Character-Set"`
 This specifies the MIME type of the output. The default is `text/html`. For example, the directive

```
<%@ page contentType="text/plain" %>
```

 has the same effect as the scriptlet

```
<% response.setContentType("text/plain"); %>
```
- `isThreadSafe="true|false"`. A value of `true` (the default) indicates normal servlet processing, where multiple requests can be processed simultaneously with a single servlet instance, under the assumption that the author synchronized access to instance variables. A value of `false` indicates that the servlet should implement `SingleThreadModel`, with requests either delivered serially or with simultaneous requests be given separate servlet instances.
- `session="true|false"`. A value of `true` (the default) indicates that the predefined variable `session` (of type `HttpSession`) should be bound to the existing session if one exists, otherwise a new session should be created and bound to it. A value of `false` indicates that no sessions will be used, and attempts to access the variable `session` will result in errors at the time the JSP page is translated into a servlet.
- `buffer="sizekb|none"`. This specifies the buffer size for the `JspWriter out`. The default is server specific, but must be at least 8kb.
- `autoflush="true|false"`. A value of `true`, the default, indicates that the buffer should be flushed when it is full. A value of `false`, rarely used, indicates that an exception should be thrown when the buffer overflows. A value of `false` is illegal when also using `buffer="none"`.
- `extends="package.class"`. This indicates the superclass of servlet that will be generated. Use this with extreme caution, since the server may be using a custom superclass already.
- `info="message"`. This defines a string that can be retrieved via the `getServletInfo` method.
- `errorPage="url"`. This specifies a JSP page that should process any `Throwables` thrown but not caught in the current page.
- `isErrorPage="true|false"`. This indicates whether or not the current page can act as the error page for another JSP page. The default is `false`.

- `language="java"`. At some point, this is intended to specify the underlying language being used. For now don't bother with this since `java` is both the default and the only legal choice.

The XML syntax for defining directives is

```
<jsp:directive.directiveType attribute=value />
```

For example, the XML equivalent of

```
<%@ page import="java.util.*" %>
```

is

```
<jsp:directive.page import="java.util.*" />
```

5.2 The JSP `include` Directive

This directive lets you include files at the time the JSP page is translated into a servlet. The directive looks like this:

```
<%@ include file="relative url" %>
```

The URL specified is normally interpreted relative to the JSP page that refers to it, but, as with relative URLs in general, you can tell the system to interpret the URL relative to the home directory of the Web server by starting the URL with a forward slash. The contents of the included file are parsed as regular JSP text, and thus can include static HTML, scripting elements, directives, and actions.

For example, many sites include a small navigation bar on each page. Due to problems with HTML frames, this is usually implemented by way of a small table across the top of the page or down the left-hand side, with the HTML repeated for each page in the site. The `include` directive is a natural way of doing this, saving the developers from the maintenance nightmare of actually copying the HTML into each separate file. Here's some representative code:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Servlet Tutorial: JavaServer Pages (JSP) 1.0</TITLE>
<META NAME="author" CONTENT="webmaster@somesite.com">
<META NAME="keywords" CONTENT="...">
<META NAME="description" CONTENT="...">
<LINK REL=STYLESHEET
      HREF="Site-Styles.css"
      TYPE="text/css">
</HEAD>

<BODY>
<%@ include file="/navbar.html" %>

<!-- Part specific to this page ... -->

</BODY>
</HTML>
```

Note that since the `include` directive inserts the files at the time the page is translated, if the navigation bar changes you need to re-translate all the JSP pages that refer to it. This is a good compromise in a situation like this, since the navigation bar probably changes infrequently, and you want the inclusion process to be as efficient as possible. If, however, the included files changed more often, you could use the `jsp:include` action instead. This includes the file at the time the JSP page is requested, and is discussed in [the tutorial section on JSP actions](#).

6. Example Using Scripting Elements and Directives

Here is a simple example showing the use of JSP expressions, scriptlets, declarations, and directives. You can also [download the source](#) or [try it on-line](#).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Using JavaServer Pages</TITLE>

<META NAME="author" CONTENT="Marty Hall -- hall@apl.jhu.edu">
<META NAME="keywords"
      CONTENT="JSP,JavaServer Pages,servlets">
<META NAME="description"
      CONTENT="A quick example of the four main JSP tags.">
<LINK REL=STYLESHEET
      HREF="My-Style-Sheet.css"
      TYPE="text/css">
</HEAD>

<BODY BGCOLOR="#FDF5E6" TEXT="#000000" LINK="#0000EE"
      VLINK="#551A8B" ALINK="#FF0000">

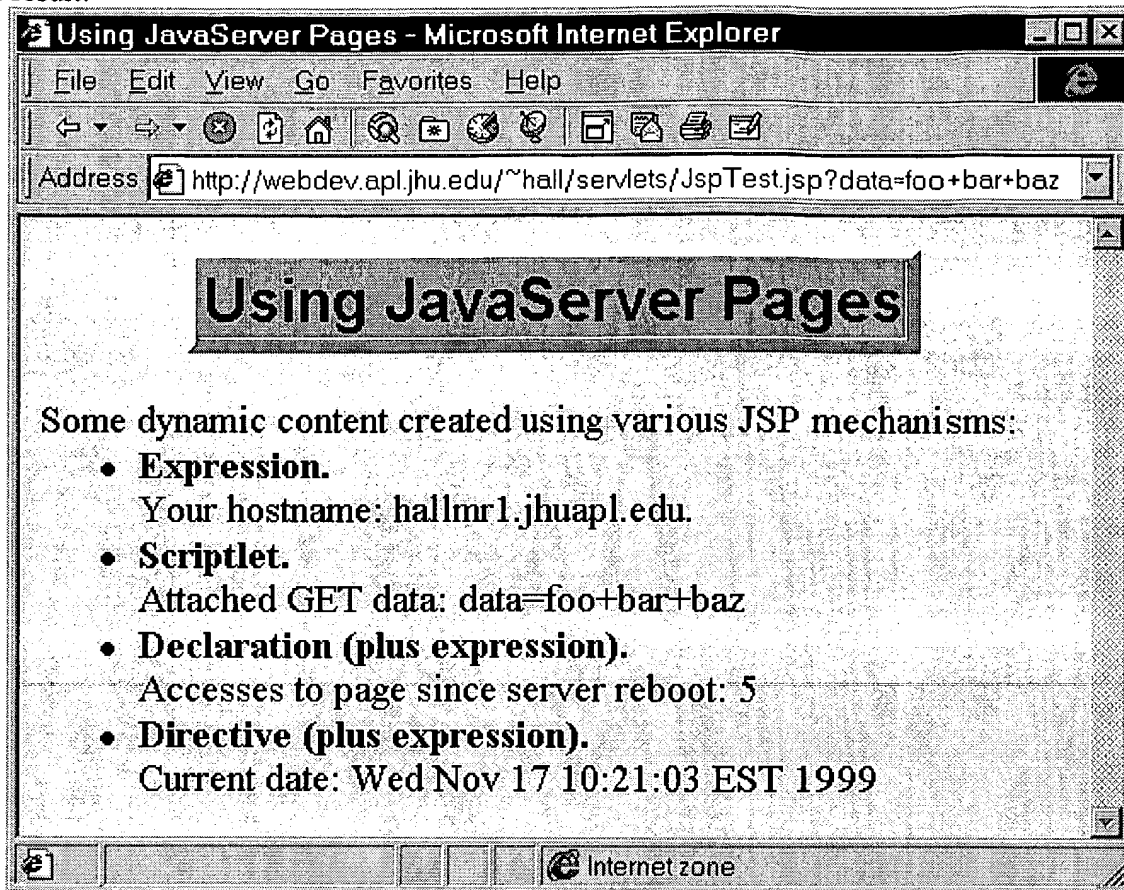
<CENTER>
<TABLE BORDER=5 BGCOLOR="#EF8429">
  <TR><TH CLASS="TITLE">
    Using JavaServer Pages</TH></TR></TABLE>
</CENTER>
<P>
```

Some dynamic content created using various JSP mechanisms:

```
<UL>
  <LI><B>Expression.</B><BR>
    Your hostname: <%= request.getRemoteHost() %>.
  <LI><B>Scriptlet.</B><BR>
    <% out.println("Attached GET data: " +
                    request.getQueryString()); %>
  <LI><B>Declaration (plus expression).</B><BR>
    <%! private int accessCount = 0; %>
    Accesses to page since server reboot: <%= ++accessCount %>
  <LI><B>Directive (plus expression).</B><BR>
    <%@ page import = "java.util.*" %>
    Current date: <%= new Date() %>
</UL>
```

```
</BODY>
</HTML>
```

Here's a typical result:



7. Predefined Variables

To simplify code in JSP expressions and scriptlets, you are supplied with eight automatically defined variables, sometimes called *implicit objects*. The available variables are `request`, `response`, `out`, `session`, `application`, `config`, `pageContext`, and `page`. Details for each are given below.

7.1 request

This is the `HttpServletRequest` associated with the request, and lets you look at the request parameters (via `getParameter`), the request type (GET, POST, HEAD, etc.), and the incoming HTTP headers (cookies, Refere etc.). Strictly speaking, `request` is allowed to be a subclass of `ServletRequest` other than `HttpServletRequest`, if the protocol in the request is something other than HTTP. This is almost never done practice.

7.2 response

This is the `HttpServletResponse` associated with the response to the client. Note that, since the output stream (see `out` below) is buffered, it is legal to set HTTP status codes and response headers, even though this is not permitted in regular servlets once any output has been sent to the client.

7.3 out

This is the `PrintWriter` used to send output to the client. However, in order to make the `response` object (see the previous section) useful, this is a buffered version of `PrintWriter` called `JspWriter`. Note that you can adjust the buffer size, or even turn buffering off, through use of the `buffer` attribute of the `page` directive. This was discussed in [Section 5](#). Also note that `out` is used almost exclusively in scriptlets, since JSP expressions automatically get placed in the output stream, and thus rarely need to refer to `out` explicitly.

7.4 session

This is the `HttpSession` object associated with the request. Recall that sessions are created automatically, so this variable is bound even if there was no incoming session reference. The one exception is if you use the `session` attribute of the `page` directive (see [Section 5](#)) to turn sessions off, in which case attempts to reference the session variable cause errors at the time the JSP page is translated into a servlet.

7.5 application

This is the `ServletContext` as obtained via `getServletConfig().getContext()`.

7.6 config

This is the `ServletConfig` object for this page.

7.7 pageContext

JSP introduced a new class called `PageContext` to encapsulate use of server-specific features like higher performance `JspWriters`. The idea is that, if you access them through this class rather than directly, your code will still run on "regular" servlet/JSP engines.

7.8 page

This is simply a synonym for `this`, and is not very useful in Java. It was created as a placeholder for the time when the scripting language could be something other than Java.

8. Actions

JSP *actions* use constructs in XML syntax to control the behavior of the servlet engine. You can dynamically insert a file, reuse JavaBeans components, forward the user to another page, or generate HTML for the Java plugin. Available actions include:

- `jsp:include` - Include a file at the time the page is requested. See [Section 8.1](#).
- `jsp:useBean` - Find or instantiate a JavaBean. See [Section 8.2](#) for an overview, and [Section 8.3](#) for details.
- `jsp:setProperty` - Set the property of a JavaBean. See [Section 8.4](#).
- `jsp:getProperty` - Insert the property of a JavaBean into the output. See [Section 8.5](#).
- `jsp:forward` - Forward the requester to a new page. See [Section 8.6](#).
- `jsp:plugin` - Generate browser-specific code that makes an `OBJECT` or `EMBED` tag for the Java plugin. See [Section 8.7](#).

These actions are described in more detail below. Remember that, as with XML in general, the element and attribute names are case sensitive.

8.1 The `jsp:include` Action

This action lets you insert files into the page being generated. The syntax looks like this:

```
<jsp:include page="relative URL" flush="true" />
```

Unlike the `include` directive, which inserts the file at the time the JSP page is translated into a servlet, this action inserts the file at the time the page is requested. This pays a small penalty in efficiency, and precludes the included page from containing general JSP code (it cannot set HTTP headers, for example), but it gains significantly in flexibility. For example, here is a JSP page that inserts four different snippets into a "What's New?" Web page. Each time the headlines change, authors only need to update the four files, but can leave the main JSP page unchanged.

WhatsNew.jsp

You can also [download the source](#) or [try it on-line](#).

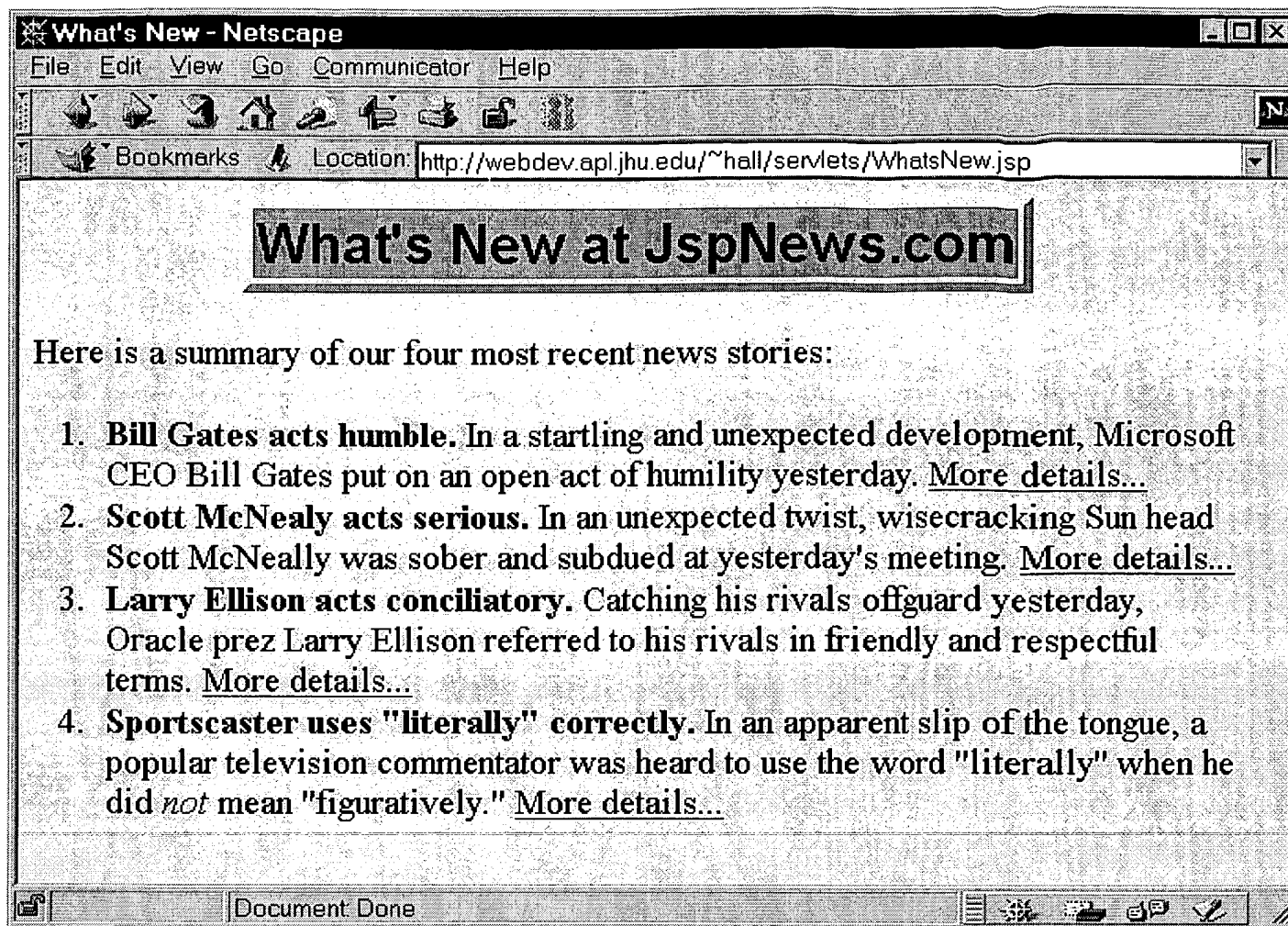
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>What's New</TITLE>
<LINK REL=STYLESHEET
      HREF="My-Style-Sheet.css"
      TYPE="text/css">
</HEAD>

<BODY BGCOLOR="#FDF5E6" TEXT="#000000" LINK="#0000EE"
      VLINK="#551A8B" ALINK="#FF0000">

<CENTER>
<TABLE BORDER=5 BGCOLOR="#EF8429">
  <TR><TH CLASS="TITLE">
    What's New at JspNews.com</TH></TR></TABLE>
</CENTER>
<P>

Here is a summary of our four most recent news stories:
<OL>
  <LI><jsp:include page="news/Item1.html" flush="true"/>
  <LI><jsp:include page="news/Item2.html" flush="true"/>
  <LI><jsp:include page="news/Item3.html" flush="true"/>
  <LI><jsp:include page="news/Item4.html" flush="true"/>
</OL>
</BODY>
</HTML>
```

Here's a typical result:



8.2 The `jsp:useBean` Action

This action lets you load in a JavaBean to be used in the JSP page. This is a very useful capability because it lets you exploit the reusability of Java classes without sacrificing the convenience that JSP adds over servlets alone. The simplest syntax for specifying that a bean should be used is:

```
<jsp:useBean id="name" class="package.class" />
```

This usually means "instantiate an object of the class specified by `class`, and bind it to a variable with the name specified by `id`." However, as we'll see shortly, you can specify a `scope` attribute that makes the bean associated with more than just the current page. In that case, it is useful to obtain references to existing beans, and the `jsp:useBean` action specifies that a new object is instantiated only if there is no existing one with the same `id` and `scope`. Now, once you have a bean, you can modify its properties via `jsp:setProperty`, or by using a scriptlet and calling a method explicitly on the object with the variable name specified earlier via the `id` attribute. Recall that with beans, when you say "this bean has a property of type `X` called `foo`", you really mean "this class has a method called `getFoo` that returns something of type `X`, and another method called `setFoo` that takes an `X` as an argument." The `jsp:setProperty` action is discussed in more detail in the next section, but for now note that you can either supply an explicit value, give a `param` attribute to say that the value is derived from the named request parameter or just list the property to indicate that the value should be derived from the request parameter with the same name as the property. You read existing properties in a JSP expression or scriptlet by calling the appropriate `getXxx` method or more commonly, by using the `jsp:getProperty` action.

Note that the class specified for the bean must be in the server's *regular* class path, not the part reserved for classes th

get automatically reloaded when they change. For example, in the Java Web Server, it and all the classes it uses should go in the `classes` directory or be in a jar file in the `lib` directory, not be in the `servlets` directory.

Here is a very simple example that loads a bean and sets/gets a simple `String` parameter.

BeanTest.jsp

You can also [download the source](#) or [try it on-line](#).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Reusing JavaBeans in JSP</TITLE>
<LINK REL=STYLESHEET
      HREF="My-Style-Sheet.css"
      TYPE="text/css">
</HEAD>

<BODY>

<CENTER>
<TABLE BORDER=5>
  <TR><TH CLASS="TITLE">
    Reusing JavaBeans in JSP</TH></TR></TABLE>
</CENTER>
<P>

<jsp:useBean id="test" class="hall.SimpleBean" />
<jsp:setProperty name="test"
                  property="message"
                  value="Hello WWW" />

<H1>Message: <I>
<jsp:getProperty name="test" property="message" />
</I></H1>

</BODY>
</HTML>
```

SimpleBean.java

Here's the source code for the bean used in the BeanTest JSP page. You can also [download the source](#).

```
package hall;

public class SimpleBean {
  private String message = "No message specified";

  public String getMessage() {
    return(message);
  }
}
```

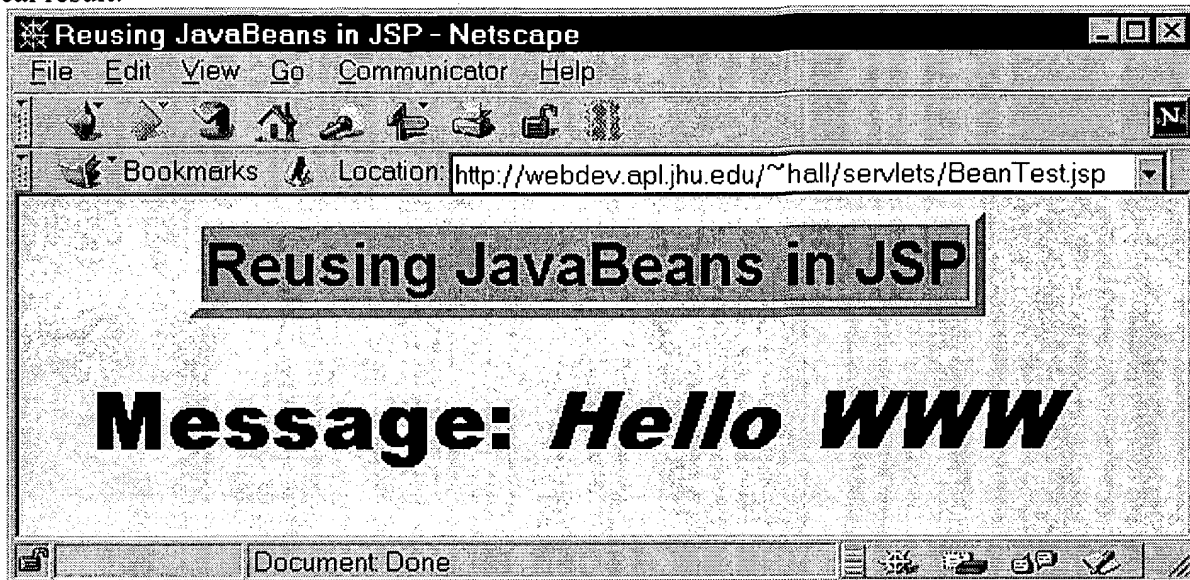


```

public void setMessage(String message) {
    this.message = message;
}
}

```

Here's a typical result:



8.3 More `jsp:useBean` Details

The simplest way to use a bean is to use

```
<jsp:useBean id="name" class="package.class" />
```

to load the bean, then use `jsp:setProperty` and `jsp:getProperty` to modify and retrieve bean properties. However, there are two other options. First, you can use the container format, namely

```
<jsp:useBean ...>
```

Body

```
</jsp:useBean>
```

to indicate that the *Body* portion should be executed only when the bean is first instantiated, not when an existing bean is found and used. As discussed below, beans can be shared, so not all `jsp:useBean` statements result in a new bean being instantiated. Second, in addition to `id` and `class`, there are three other attributes that you can use: `scope`, `type`, and `beanName`. These attributes are summarized in the following table.

Attribute	Usage
<code>id</code>	Gives a name to the variable that will reference the bean. A previous bean object is used instead of instantiating a new one if one can be found with the same <code>id</code> and <code>scope</code> .
<code>class</code>	Designates the full package name of the bean.
<code>scope</code>	Indicates the context in which the bean should be made available. There are four possible values: <code>page</code> request, <code>session</code> , and <code>application</code> . The default, <code>page</code> , indicates that the bean is only available on the current page (stored in the <code>PageContext</code> of the current page). A value of <code>request</code> indicates that the bean is only available for the current client request (stored in the <code>ServletRequest</code> object). A value of <code>session</code> indicates that the object is available to all pages during the life of the current <code>HttpSession</code> . Finally, a value of <code>application</code> indicates that it is available to all pages that share the same <code>ServletContext</code> . The reason that the <code>scope</code> matters is that a <code>jsp:useBean</code> entry will only result in a new object being instantiated if there is no previous object with the same <code>id</code> .

	and scope. Otherwise the previously existing object is used, and any <code>jsp:setParameter</code> elements or other entries between the <code>jsp:useBean</code> start and end tags will be ignored.
type	Specifies the type of the variable that will refer to the object. This must match the classname or be a superclass or an interface that the class implements. Remember that the <i>name</i> of the variable is designated via the <code>id</code> attribute.
beanName	Gives the name of the bean, as you would supply it to the <code>instantiate</code> method of Beans. It is permissible to supply a type and a beanName, and omit the <code>class</code> attribute.

8.4 The `jsp:setProperty` Action

You use `jsp:setProperty` to give values to properties of beans that have been referenced earlier. You can do it in two contexts. First, you can use `jsp:setProperty` after, but outside of, a `jsp:useBean` element, as below

```
<jsp:useBean id="myName" ... />
...
<jsp:setProperty name="myName"
    property="someProperty" ... />
```

In this case, the `jsp:setProperty` is executed regardless of whether a new bean was instantiated or an existing bean was found. A second context in which `jsp:setProperty` can appear is inside the body of a `jsp:useBean` element, as below:

```
<jsp:useBean id="myName" ... >
    ...
    <jsp:setProperty name="myName"
        property="someProperty" ... />
</jsp:useBean>
```

Here, the `jsp:setProperty` is executed only if a new object was instantiated, not if an existing one was found.

There are four possible attributes of `jsp:setProperty`:

Attribute	Usage
name	This required attribute designates the bean whose property will be set. The <code>jsp:useBean</code> element must appear before the <code>jsp:setProperty</code> element.
property	This required attribute indicates the property you want to set. However, there is one special case: a value of "*" means that all request parameters whose names match bean property names will be passed to the appropriate setter methods.
value	This optional attribute specifies the value for the property. String values are automatically converted to numbers, boolean, Boolean, byte, Byte, char, and Character via the standard <code>valueOf</code> method in the target or wrapper class. For example, a value of "true" for a boolean or Boolean property will be converted via <code>Boolean.valueOf</code> , and a value of "42" for an int or Integer property will be converted via <code>Integer.valueOf</code> . You can't use both <code>value</code> and <code>param</code> , but it is permissible to use neither. See the discussion of <code>param</code> below.
	This optional attribute designates the request parameter from which the property should be derived. If the current request has no such parameter, nothing is done: the system does <i>not</i> pass null to the setter method of the property. Thus, you can let the bean itself supply default values, overriding them only when the request parameters say to do so. For example, the following snippet says "set the

numberOfItems property to whatever the value of the numItems request parameter is, if there is such a request parameter. Otherwise don't do anything."

```
<jsp:setProperty name="orderBean"
                 property="numberOfItems"
                 param="numItems" />
```

param

If you omit both value and param, it is the same as if you supplied a param name that matches the property name. You can take this idea of automatically using the request property whose name matches the property one step further by supplying a property name of "*" and omitting both value and param. In this case, the server iterates through available properties and request parameters, matching up ones with identical names.

Here's an example that uses a bean to create a table of prime numbers. If there is a parameter named numDigits in the request data, it is passed into the bean's numDigits property. Likewise for numPrimes.

JspPrimes.jsp

To download the JSP source, right click on [the source code link](#). You can also download [the source code for the NumberedPrimes bean](#) referenced by the jsp:useBean element. Browse the [source code directory](#) for other Java classes used by NumberedPrimes. The best way to try it out on-line is to start with [the HTML page that act as a front end to it](#).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Reusing JavaBeans in JSP</TITLE>
<LINK REL=STYLESHEET
      HREF="My-Style-Sheet.css"
      TYPE="text/css">
</HEAD>

<BODY>

<CENTER>
<TABLE BORDER=5>
  <TR><TH CLASS="TITLE">
    Reusing JavaBeans in JSP</TH></TR></TABLE>
</CENTER>
<P>
```

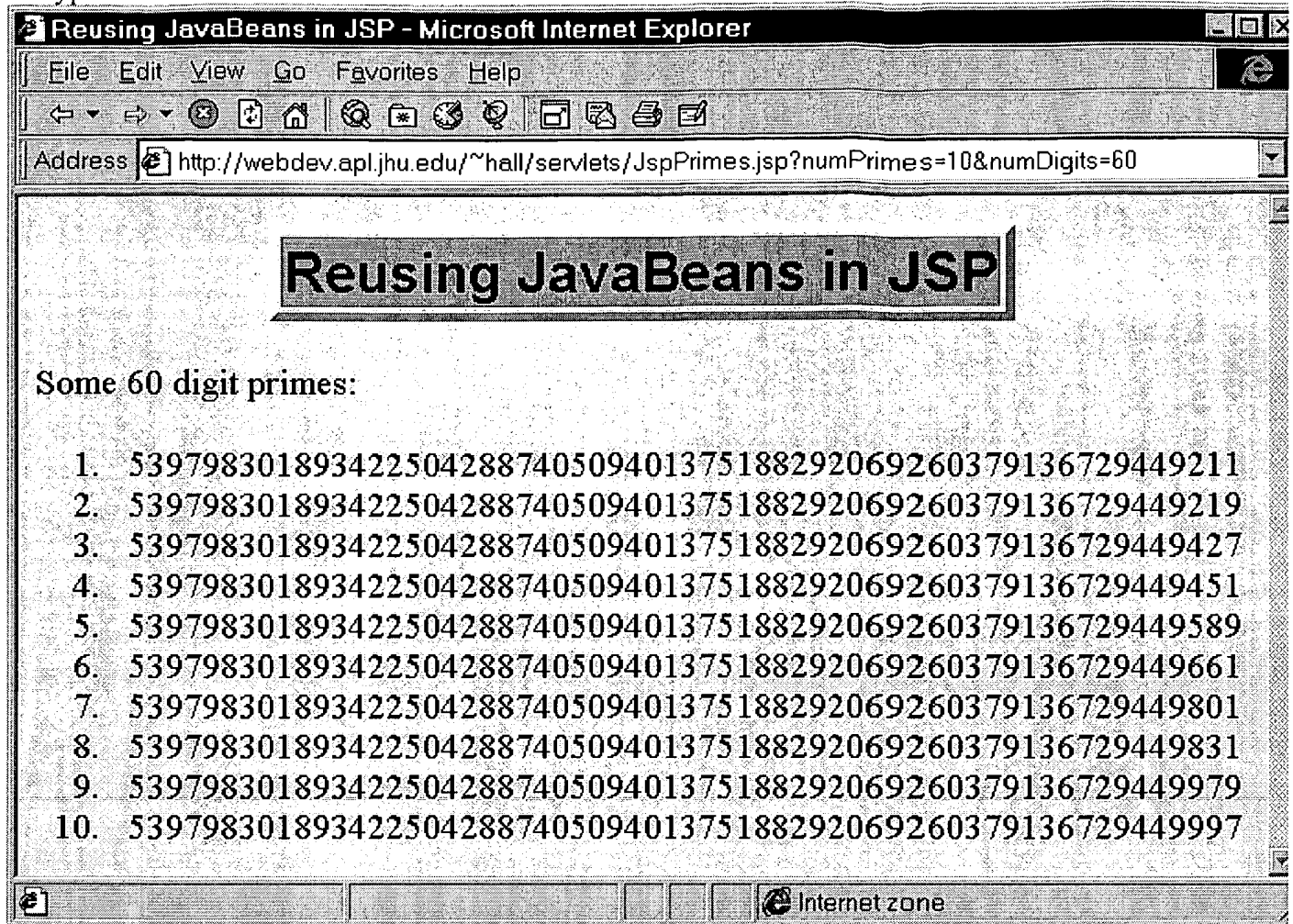
```
<jsp:useBean id="primeTable" class="hall.NumberedPrimes" />
<jsp:setProperty name="primeTable" property="numDigits" />
<jsp:setProperty name="primeTable" property="numPrimes" />
```

Some <jsp:getProperty name="primeTable" property="numDigits" />
digit primes:

```
<jsp:getProperty name="primeTable" property="numberedList" />
```

```
</BODY>
</HTML>
```

Here's a typical result:



8.5 The `jsp:getProperty` Action

This element retrieves the value of a bean property, converts it to a string, and inserts it into the output. The two required attributes are `name`, the name of a bean previously referenced via `jsp:useBean`, and `property`, the property whose value should be inserted. Here's an example; for more examples, see Sections [8.2](#) and [8.4](#).

```
<jsp:useBean id="itemBean" ... />
...
<UL>
  <LI>Number of items:
    <jsp:getProperty name="itemBean" property="numItems" />
  <LI>Cost of each:
    <jsp:getProperty name="itemBean" property="unitCost" />
</UL>
```

8.6 The `jsp:forward` Action

This action lets you forward the request to another page. It has a single attribute, `page`, which should consist of a relative URL. This could be a static value, or could be computed at request time, as in the two examples below.

```
<jsp:forward page="/utils/errorReporter.jsp" />
<jsp:forward page="<%= someJavaExpression %>" />
```

8.7 The `jsp:plugin` Action

This action lets you insert the browser-specific OBJECT or EMBED element needed to specify that the browser run a applet using the Java plugin.

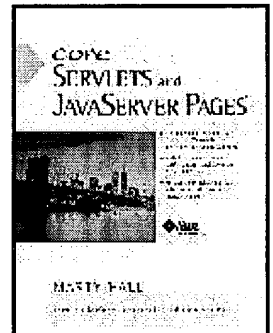
9. Comments and Character Quoting Conventions

There are a small number of special constructs you can use in various cases to insert comments or characters that would otherwise be treated specially. Here's a summary:

Syntax	Purpose
<code><%-- comment --%></code>	A JSP comment. Ignored by JSP-to-scriptlet translator. Any embedded JSP scripting elements, directives, or actions are ignored.
<code><!-- comment --></code>	An HTML comment. Passed through to resultant HTML. Any embedded JSP scripting elements, directives, or actions are executed normally.
<code><\%</code>	Used in template text (static HTML) where you really want "<%".
<code>%\></code>	Used in scripting elements where you really want "%>".
<code>\'</code>	A single quote in an attribute that uses single quotes. Remember, however, that you can use either single or double quotes, and the other type of quote will then be a regular character.
<code>\"</code>	A double quote in an attribute that uses double quotes. Remember, however, that you can use either single or double quotes, and the other type of quote will then be a regular character.
<code>%\></code>	<code>%></code> in an attribute.
<code><\%</code>	<code><%</code> in an attribute.

This page is part of my [Tutorial on Servlets and JSP](http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-JSP.html). © 1999 [Marty Hall](http://www.apl.jhu.edu/~hall/). All source code freely available for unrestricted use.

This tutorial was a very early outline of the material that eventually became *Core Servlets and JavaServer Pages* (Sun Microsystems Press 2000). See <http://www.coreservlets.com/> for the topics covered, table of contents, index, reader reviews, and complete source code archive.



Order now

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet

Your search matched **61** of **1074479** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or entering a new one in the text box.

java <near> server <near> page

☐ Check to search within this result set

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 Teachers Assistant: a Web-based application using Java Server Pages and Java database connectivity

Barua, S.;

Systems, Man, and Cybernetics, 2000 IEEE International Conference on , Volume: 2 , 8-11 Oct. 2000

Pages:828 - 833 vol.2

[\[Abstract\]](#) [\[PDF Full-Text \(504 KB\)\]](#) **IEEE CNF**

2 An evolvable software framework for an Internet-based telediagnostic system

Mitsa, T.;

Information Technology Applications in Biomedicine, 2003. 4th International IEEE EMBS Special Topic Conference on , 24-26 April 2003

Pages:203 - 206

[\[Abstract\]](#) [\[PDF Full-Text \(444 KB\)\]](#) **IEEE CNF**

3 Migrating COBOL systems to the Web by using the MVC design pattern

Bodhuin, T.; *Guardabascio, E.;* *Tortorella, M.;*

Reverse Engineering, 2002. Proceedings. Ninth Working Conference on , 29 Oct.-1 Nov. 2002

Pages:329 - 338

[\[Abstract\]](#) [\[PDF Full-Text \(556 KB\)\]](#) **IEEE CNF**

4 Re-engineering of E-commerce legacy applications for Enterprise Java/spl trade/ environment

Lau, T.C.; *Chan, Y.;* *Khuwaja, R.;* *Jianguo Lu;*

Software Maintenance, 2002. Proceedings. International Conference on , 3-6 Oct. 2002

Pages:573

[\[Abstract\]](#) [\[PDF Full-Text \(206 KB\)\]](#) **IEEE CNF**

5 Automatic face recognition system architecture for collaborative virtual environments

Adler, A.;

Haptic Virtual Environments and Their Applications, IEEE International Workshop 2002 HAVE , 17-18 Nov. 2002

Pages:1 - 6

[[Abstract](#)] [[PDF Full-Text \(558 KB\)](#)] IEEE CNF

6 Multi-tiered Internet computing using Java technologies

Liu, J.B.;

Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE , Volume: 3 , 29 Nov.-2 Dec. 2001

Pages:1789 - 1793 vol.3

[[Abstract](#)] [[PDF Full-Text \(713 KB\)](#)] IEEE CNF

7 A Java-based method for developing Web application system

Li Chunlin;

Communications, 1999. APCC/OECC '99. Fifth Asia-Pacific Conference on ... and Fourth Optoelectronics and Communications Conference , Volume: 2 , 18-22 Oct. 1999

Pages:1079 - 1082 vol.2

[[Abstract](#)] [[PDF Full-Text \(348 KB\)](#)] IEEE CNF

8 Product development process capture and display using Web-based technologies

Sabbaghian, N.; Eppinger, S.; Murman, E.;

Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on , Volume: 3 , 11-14 Oct. 1998

Pages:2664 - 2669 vol.3

[[Abstract](#)] [[PDF Full-Text \(680 KB\)](#)] IEEE CNF

9 Development of distance learning environment

Khan, S.A.; Devgan, S.S.;

Southeastcon 2000. Proceedings of the IEEE , 7-9 April 2000

Pages:519 - 522

[[Abstract](#)] [[PDF Full-Text \(264 KB\)](#)] IEEE CNF

10 A comparison of two object oriented application development platforms for multimedia services to the Internet

Michalas, A.; Zoi, V.; Sotiropoulos, N.; Mitrou, N.; Loumos, V.; Kayafas, E.;

Electrotechnical Conference, 2000. MELECON 2000. 10th Mediterranean , Volume: 2 , 2000

Pages:750 - 753 vol.2

[[Abstract](#)] [[PDF Full-Text \(284 KB\)](#)] IEEE CNF

11 An integrated system based on Web and/or WAP framework for remote monitoring and control of industrial processes

Nikolakopoulos, G.; Koundourakis, M.; Tzes, A.;

Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS '03. 2003 IEEE International Symposium on , 27-29 July 2003

Pages:201 - 206

[\[Abstract\]](#) [\[PDF Full-Text \(484 KB\)\]](#) [IEEE CNF](#)

12 Selective Java applet filtering on Internet

Chen Ding; Jing Deng; Chi-Hung Chi; Chun-Lei Dong;

Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999

IEEE International Conference on , Volume: 2 , 12-15 Oct. 1999

Pages:110 - 114 vol.2

[\[Abstract\]](#) [\[PDF Full-Text \(380 KB\)\]](#) [IEEE CNF](#)

13 Thread-aware garbage collection for server applications

Woo Jin Kim; Kyungbaek Kim; Han, J.; Park, K.; Park, D.;

Applications and the Internet, 2004. Proceedings. 2004 International Symposium on , 2004

Pages:81 - 87

[\[Abstract\]](#) [\[PDF Full-Text \(282 KB\)\]](#) [IEEE CNF](#)

14 Hands on software engineering: building a Java software engineering development environment

Shafer, D.;

Frontiers in Education Conference, 2001. 31st Annual , Volume: 2 , 10-13 Oct. 2001

Pages:F1D - 13-19 vol.2

[\[Abstract\]](#) [\[PDF Full-Text \(474 KB\)\]](#) [IEEE CNF](#)

15 PREORD-a prototype of an object oriented Web browser based DMS system

Silva, M.P.; Saraiva, J.T.; Sousa, A.V.;

Power Tech Proceedings, 2001 IEEE Porto , Volume: 3 , 10-13 Sept. 2001

Pages:6 pp. vol.3

[\[Abstract\]](#) [\[PDF Full-Text \(613 KB\)\]](#) [IEEE CNF](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)
